



# MCL-Link Version 2.10



## User's Guide

**MCL-Link Version 2.10 User's Guide**



70-33346-03  
Revision .1 — December 1999

***MCL-Link Version 2.10***  
***User's Guide***

70-33346-03

*Revision .1*

*December 1999*

Preliminary



© 1998-1999 by Symbol Technologies, Inc. All rights reserved.

No part of this publication may be reproduced or used in any form, or by any electrical or mechanical means, without permission in writing from Symbol. This includes electronic or mechanical means, such as photocopying, recording, or information storage and retrieval systems. The material in this manual is subject to change without notice.

The software is provided strictly on an “as is” basis. All software, including firmware, furnished to the user is on a licensed basis. Symbol grants to the user a non-transferable and non-exclusive license to use each software or firmware program delivered hereunder (licensed program). Except as noted below, such license may not be assigned, sublicensed, or otherwise transferred by the user without prior written consent of Symbol. No right to copy a licensed program in whole or in part is granted, except as permitted under copyright law. The user shall not modify, merge, or incorporate any form or portion of a licensed program with other program material, create a derivative work from a licensed program, or use a licensed program in a network without written permission from Symbol. The user agrees to maintain Symbol’s copyright notice on the licensed programs delivered hereunder, and to include the same on any authorized copies it makes, in whole or in part. The user agrees not to decompile, disassemble, decode, or reverse engineer any licensed program delivered to the user or any portion thereof.

Symbol reserves the right to make changes to any software or product to improve reliability, function, or design.

Symbol does not assume any product liability arising out of, or in connection with, the application or use of any product, circuit, or application described herein.

No license is granted, either expressly or by implication, estoppel, or otherwise under any Symbol Technologies, Inc., intellectual property rights. An implied license only exists for equipment, circuits, and subsystems contained in Symbol products.

Symbol, Spectrum One, and Spectrum24 are registered trademarks of Symbol Technologies, Inc. MCL, MCL-Designer, MCL-Link, MCL-Loader, and MCL-Net S24 are trademarks of Zetes Technologies S.A. Other product names mentioned in this manual may be trademarks or registered trademarks of their respective companies and are hereby acknowledged.

Symbol Technologies, Inc.  
One Symbol Plaza  
Holtsville, New York 11742-1300  
<http://www.symbol.com>



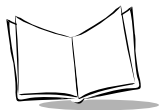
# Contents

## About This Guide

Notational Conventions .....	vii
Related Publications .....	viii
Service Information .....	viii
Symbol Support Centers .....	ix

## Chapter I. Getting Started

Introduction .....	1-1
Installation .....	1-2
MCL-Link Operating Modes .....	1-3
Starting MCL-Link .....	1-3
Starting MCL-Link IT - Single Instance .....	1-4
Starting the MCL-Link IT - Multi-Instance .....	1-5
Starting MCL-Link with Arguments .....	1-7
Communication Setup .....	1-8
Protocol .....	1-9
ODBC .....	1-10
Dx Packet .....	1-12
Interface .....	1-13
Terminal .....	1-14
Using MCL-Link .....	1-16
Status .....	1-16
Receive .....	1-17
Send .....	1-18
Script .....	1-19
Time .....	1-20
About MCL-Link Files .....	1-20
MCL-Link Configuration File .....	1-21
MCL Link Error File .....	1-27
Troubleshooting .....	1-30



## Chapter 2. Command File

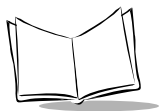
Introduction.....	2-1
Syntax Of Commands .....	2-3
Remote Commands .....	2-4
Transmit Program (TM).....	2-4
Transmit Data File (TF) .....	2-5
Data File Request (FR) .....	2-6
Transmit Current Date and Time (TT) .....	2-7
Reset Terminal (RZ).....	2-8
Status Request (SR) .....	2-9
File Copy (FC) .....	2-10
File Rename (FN) .....	2-11
File Delete (FD) .....	2-12
File Append (FA) .....	2-13
Exit MCL-Link (QX) .....	2-14
Local Commands.....	2-15
Wait (WT) .....	2-15
Quit (QX).....	2-16
Quit on Time Out (QT) .....	2-17
Label (LB).....	2-18
Skip (SK) .....	2-19
Test And Branch (IF) .....	2-20
Modem Commands (MD) .....	2-21
Comment (**) .....	2-22
Local File Copy (FC) .....	2-23
Local File Rename (FN) .....	2-24
Local File Delete (FD).....	2-25
Local File Append (FA).....	2-26
Execute (EX) .....	2-27
Commands Received from the Terminal .....	2-28
Receive Data Packet (D0-D9) from terminal.....	2-28
File Look-Up (CR) from terminal .....	2-29
Transmit Remote Data File (TF) from terminal .....	2-30
Remote Data File Request (FR) from terminal .....	2-31
Host Status Request (SR) from terminal.....	2-32
Host File Copy (FC) from terminal.....	2-33
Host File Rename (FN) from terminal .....	2-34
Host File Delete (FD) from terminal .....	2-35
Host File Append (FA) from terminal.....	2-36
Quit MCL-Link (QX) from terminal .....	2-37
Execute a Program (EX) from terminal.....	2-38
MCL-Link DLL.....	2-39
DLL Conventions.....	2-39

Return Code Standard Values .....	2-40
MCL-Link DLL Functions .....	2-41
MCLLink_Start Function .....	2-41
MCLLink_Stop Function .....	2-42
MCLLink_Check Function .....	2-43
MCLLink_OpenPipe Function .....	2-44
MCLLink_ClosePipe Function .....	2-45
MCLLink_WaitData Function .....	2-46
MCLLink_SendData Function .....	2-48
MCLLink_CheckTerminal Function .....	2-50

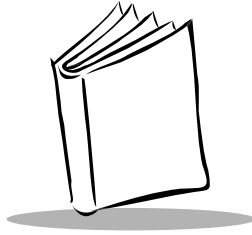
## Index

## Feedback

Preliminary



Preliminary



## About This Guide

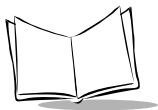
The *MCL-Link Version 2.10 User's Guide* provides general information about operating the MCL-Link application, configuring the software, and using MCL-Link commands.

### Notational Conventions

---

The following conventions are used in this document:

- ◆ “Operator” and “User” refer to anyone using the MCL-Link software.
- ◆ “PC” refers to the IBM personal computer or compatible system that you are using to develop applications.
- ◆ “Terminal” refers to various types of Symbol terminals.
- ◆ “You” refers to the administrator or person who is using this guide as a reference aid to install, configure, and/or operate the software.
- ◆ Keystrokes in bold type indicate non-alphanumeric keystrokes. For example: Select the **<F1>** key on the terminal to access on-line help.
- ◆ **Bold** type identifies menu items and input or text fields on a terminal screen.
- ◆ *Italics* are used:
  - ◆ for the names of parameters in function prototypes and variable names in usage and syntax descriptions
  - ◆ to highlight specific items in the general text
  - ◆ to identify chapters and sections in this and related documents.
- ◆ Square brackets [ ] in a command line enclose optional command line parameters.
- ◆ The piping symbol | has the effect of “or” when it is used to separate inline parameters on a command line; i.e., it separates alternative values for parameters.



- ◆ Bullets (●) indicate:
  - ◆ action items
  - ◆ lists of alternatives
  - ◆ lists of required steps that are not necessarily sequential
- ◆ Sequential lists (e.g., those that describe step-by-step procedures) appear as numbered lists.

## Related Publications

---

The following is a list of documents that you may find useful if you want to know more about the MCL-Link program.

- ◆ *MCL-Loader User's Guide*  
p/n 70-33347-xx
- ◆ *MCL-Net S24 User's Guide*  
p/n 70-33348-xx
- ◆ *MCL-Designer User's Guide*  
p/n 70-33345-xx

## Service Information

---

If you have a problem with your equipment, contact the *Symbol Support Centers*. Before calling, have the model number, serial number, and several of your bar code symbols at hand.

Call the Support Center from a phone near the equipment so that the service person can try to talk you through your problem. If the equipment is found to be working properly and the problem is symbol readability, the Support Center will request samples of your bar codes for analysis at our plant.

If your problem cannot be solved over the phone, you may need to return your equipment for servicing. If that is necessary, you will be given specific directions.

---

**Note:** *Symbol Technologies is not responsible for any damages incurred during shipment if the approved shipping container is not used. Shipping the units improperly can possibly void the warranty. If the original shipping container was not kept, contact Symbol to have another sent to you.*

---

## Symbol Support Centers

For service information, warranty information or technical assistance contact or call the Symbol Support Center in:

### United States

Symbol Technologies, Inc.  
One Symbol Plaza  
Holtsville, New York 11742-1300  
1-800-653-5350

### United Kingdom

Symbol Technologies  
Symbol Place  
Winnersh Triangle, Berkshire RG41 5TP  
United Kingdom  
0800 328 2424 (Inside UK)  
+44 118 945 7529 (Outside UK)

### Australia

Symbol Technologies Pty. Ltd.  
432 St. Kilda Road  
Melbourne, Victoria 3004  
1-800-672-906 (Inside Australia)  
+61-3-9866-6044 (Outside Australia)

### Denmark

Symbol Technologies AS  
Gydevang 2,  
DK-3450 Allerød, Denmark  
7020-1718 (Inside Denmark)  
+45-7020-1718 (Outside Denmark)

### Canada

Symbol Technologies Canada, Inc.  
2540 Matheson Boulevard East  
Mississauga, Ontario, Canada L4W 4Z2  
905-629-7226

### Asia/Pacific

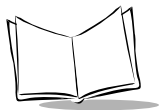
Symbol Technologies Asia, Inc.  
230 Victoria Street #04-05  
Bugis Junction Office Tower  
Singapore 188024  
337-6588 (Inside Singapore)  
+65-337-6588 (Outside Singapore)

### Austria

Symbol Technologies Austria GmbH  
Prinz-Eugen Strasse 70  
Suite 3  
2.Haus, 5.Stock  
1040 Vienna, Austria  
1-505-5794 (Inside Austria)  
+43-1-505-5794 (Outside Austria)

### Europe/Mid-East Distributor Operations

Contact your local distributor or call  
+44 118 945 7360



### **Finland**

Oy Symbol Technologies  
Kaupintie 8 A 6  
FIN-00440 Helsinki, Finland  
9 5407 580 (Inside Finland)  
+358 9 5407 580 (Outside Finland)

### **Germany**

Symbol Technologies GmbH  
Waldstrasse 68  
D-63128 Dietzenbach, Germany  
6074-49020 (Inside Germany)  
+49-6074-49020 (Outside Germany)

### **Latin America Sales Support**

7900 Glades Road  
Suite 340  
Boca Raton, Florida 33434 USA  
1-800-347-0178 (Inside United States)  
+1-561-483-1275 (Outside United States)

### **Netherlands**

Symbol Technologies  
Kerkplein 2, 7051 CX  
Postbus 24 7050 AA  
Varseveld, Netherlands  
315-271700 (Inside Netherlands)  
+31-315-271700 (Outside Netherlands)

### **France**

Symbol Technologies France  
Centre d'Affaire d'Antony  
3 Rue de la Renaissance  
92184 Antony Cedex, France  
01-40-96-52-21 (Inside France)  
+33-1-40-96-52-50 (Outside France)

### **Italy**

Symbol Technologies Italia S.R.L.  
Via Cristoforo Colombo, 49  
20090 Trezzano S/N Naviglio  
Milano, Italy  
2-484441 (Inside Italy)  
+39-02-484441 (Outside Italy)

### **Mexico**

Symbol Technologies Mexico Ltd.  
Torre Picasso  
Boulevard Manuel Avila Camacho No 88  
Lomas de Chapultepec CP 11000  
Mexico City, DF, Mexico  
5-520-1835 (Inside Mexico)  
+52-5-520-1835 (Outside Mexico)

### **Norway**

Symbol Technologies  
Trollasveien 36  
Postboks 72  
1414 Trollasen, Norway  
66810600 (Inside Norway)  
+47-66810600 (Outside Norway)

**South Africa**

Symbol Technologies Africa Inc.  
Block B2  
Rutherford Estate  
1 Scott Street  
Waverly 2090 Johannesburg  
Republic of South Africa  
11-4405668 (Inside South Africa)  
+27-11-4405668 (Outside South Africa)

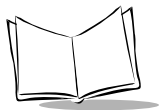
**Spain**

Symbol Technologies S.A.  
Edificio la Piovera Azul  
C. Peonias, No. 2 - Sexta Planta  
28042 Madrid, Spain  
9-1-320-39-09 (Inside Spain)  
+34-9-1-320-39-09 (Outside Spain)

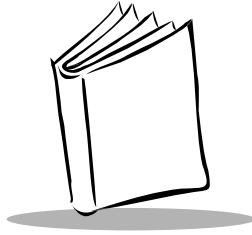
**Sweden**

Symbol Technologies AB  
Albygatan 109D  
Solna  
Sweden  
84452900 (Inside Sweden)  
+46 84452900 (Outside Sweden)

If you purchased your Symbol product from a Symbol Business Partner, contact that Business Partner for service.



Preliminary



# *Chapter 1*

## *Getting Started*

### **Introduction**

---

MCL-Link is Windows 3.1, NT/ 95 / 98 batch communication server designed to support Symbol's P460 Memory Scanner and Symbol's character-based terminals running batch applications created using MCL-designer.

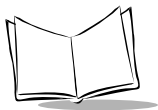
MCL-Link is the software tool that ensures access to and delivery of both programs and data essential to the smooth operation of an enterprise that relies on batch data collection.

MCL-Link communicates with your terminals and scanners either through a simple RS232 direct connection or, for remote access, using a modem. Communications can be initiated by either the server or the batch device by running MCL-Link in either a server or slave mode.

MCL-link provides the ability to handle all the complex tasks needed in a batch communications environment. Users can easily and quickly upload or download files and programs either to or from a batch device. In addition, MCL-Link enables users to:

- ◆ synchronize the terminal with the host computer
- ◆ query the terminal's status
- ◆ query terminal directories and files structures
- ◆ perform file operations, such as file lookup, save, file append, file copy, file rename, and program execution.

MCL-Link's powerful instruction set gives you the ability to create complex communications scripts with virtually no need for human intervention on either the device or the server. This is due to MCL-Link's ability to perform actions based upon instructions it receives from the device (through the serial line or modem) or from a program on the server.



MCL-Link can be started one or several times (instance) on the same PC following the type of the security hardware dongle connected onto the PC parallel port. Each instance of MCL-Link can manage one serial communication port of the PC. By this mean, it's possible to communicate with terminals through multiple communication ports on the same PC.

MCL-Link includes a SQL engine that provides the ability to access and perform SQL request on database using ODBC.

## Installation

---

MCL-Link is part of the MCL Collection of software. It can be installed separately or as part of the whole MCL Collection series. To install MCL-Link from the MCL Collection CD:

1. Insert the MCL-Collection CD-ROM disk into the CD-ROM drive.
2. Click the **Start** button on the task bar.
3. Select **Run**.
4. Type **D:\setup.exe.** and click **OK** (replace **D:** with the CD-ROM drive letter).
5. Click **Next** when the introductory screen appears.
6. Click **Yes** to accept the license agreement.
7. When prompted, type your name and company in the appropriate fields. Click **Next**.
8. Click **Next** to accept the default directory or click **Browse** to select a different directory. Click **OK**, then **Next**.
9. Click **Next** to install all the MCL programs, or select **MCL Link(16) V2.10** or **MCL Link(32) V2.10** and de-select all the other options.
10. Click **Next** to accept the MCL2 program folder.
11. Setup installs the selected programs onto the hard drive. A screen that monitors the progress appears. When installation is complete, click **Finish**.

To install MCL-Link from the MCL-Link floppy disk:

1. Insert the MCL-Link floppy disk into the floppy disk drive.
2. Click the **Start** button on the task bar.
3. Select **Run**.
4. Type **A:\setup.exe.** and click **OK** (replace **A:** with the floppy disk drive letter).
5. Click **Next** when the introductory screen appears.
6. Click **Yes** to accept the license agreement.
7. When prompted, type your name and company in the appropriate fields. Click **Next**.

8. Click **Next** to accept the default directory or click **Browse** to select a different directory. Click **OK**, then **Next**.
9. Click **Next** to install all the MCL-Link programs.
10. Click **Next** to accept the MCL program folder.
11. Setup installs the selected programs onto the hard drive. A screen that monitors the progress displays. When installation is complete, click **Finish**.

## MCL-Link Operating Modes

---

MCL-Link can receive commands:

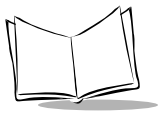
- ◆ From the script file **MCLLINK.CMD**.  
Any host application can write commands in the script file. Every second, MCL-Link checks if the script file exists, and executes the command(s) stored in it. At the end of the file, the file is renamed **MCLLINK.BAK**.  
The **MCLLINK.CMD** file must be created in the **MCLLINK.EXE** directory (See *About MCL-Link Files* on page 1-20).
- ◆ From the user buttons.  
On the supervisor's screen, buttons activate specific commands like:
  - ◆ request terminal status
  - ◆ send date and time to the terminal
  - ◆ send programs and data files to the terminal
  - ◆ request programs and data files from terminal.
- ◆ From the terminal.  
MCL-Link is able to receive commands from terminal. The communication process is controlled by the program written in the terminal. (See *Commands Received from the Terminal* on page 2-28).
- ◆ From the DLL.  
MCL-Link is able to receive commands from the DLL via named pipes. See *MCL-Link DLL* on page 2-39 for more information.

## Starting MCL-Link

---

MCL-Link IT can be started one or several times (instance) on the same PC following the type of the security hardware dongle connected onto the PC parallel port.

Each instance of the MCL-Link can manage one serial communication port of the PC.

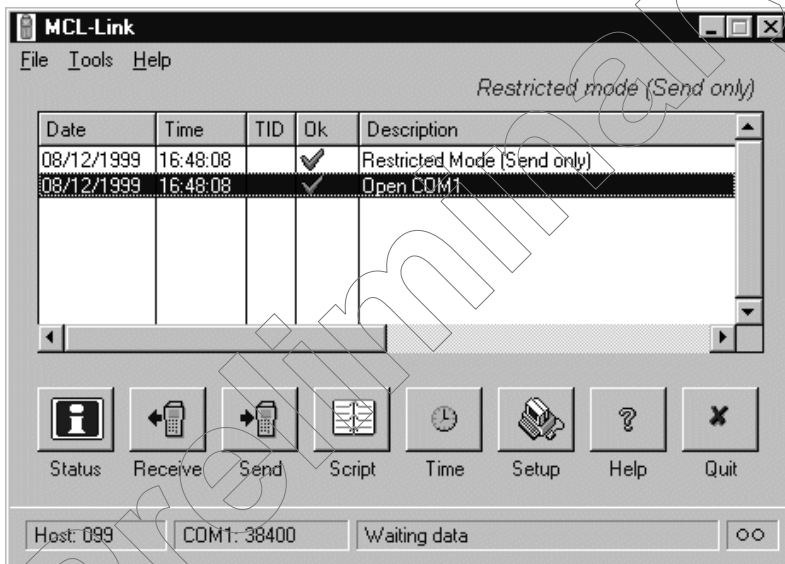


## Starting MCL-Link IT - Single Instance

There are three ways to start the MCL-Link program:

- ◆ double-clicking on the MCL-Link(16) V2.10 or MCL-Link(32) V2.10 icon on the desktop
- ◆ from the standard Run window
- ◆ from an application.

The MCL-Link window displays.



**Figure I-1. MCL-Link Window**

The list below describes the functions of the buttons on the MCL-Link window.

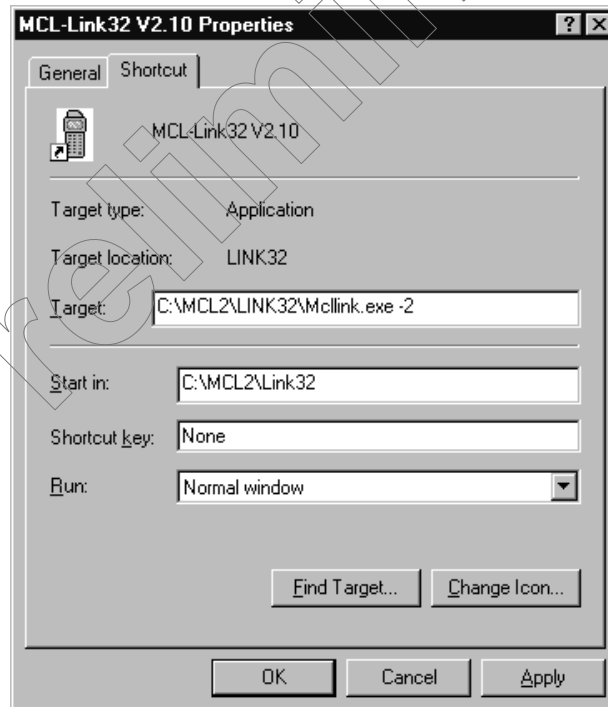
Button	Description
Status	Requests terminal status.
Receive	Receives data from the terminal.
Send	Sends data to the terminal.
Script	Associates a specified Script file to a terminal.

Button	Description
Time	Enables MCL-Link to synchronize the terminal's time.
Setup	Sets the host communication parameters.
Help	Opens the MCL-Link on-line help application.
Quit	Terminates the MCL-Link program.

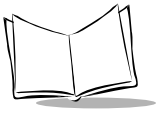
## Starting the MCL-Link IT – Multi-Instance

To define an instance of MCL-Link, create a new MCLLINK.EXE shortcut on the Windows desktop.

Click right on the new shortcut, select Properties and select shortcut tab (see Figure 1-2). Modify the Target entry to add the instance argument (-1 starts instance 1 of MCLLINK.EXE, -2 starts instance 2, ...-8 starts instance 8 of MCLLINK.EXE).



**Figure 1-2. Shortcut Tab**



By default, the argument defines the PC communication serial port that MCL-Link uses.

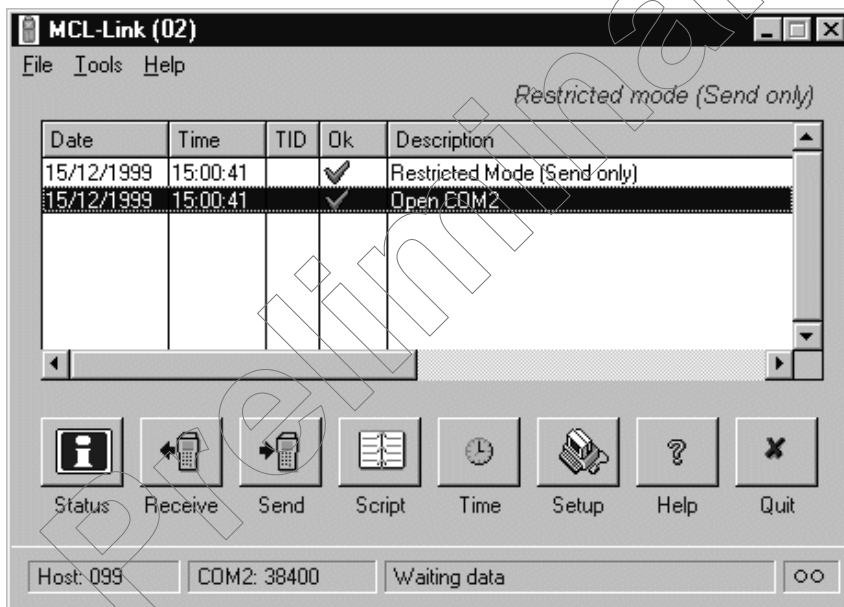
**Example:** Argument -2 = MCL-Link uses PC communication serial port 2.

The user is able to modify the com port and its settings using the setup button in the MCL-Link main window.

Each instance of MCL-Link has its own configuration file (INI) and its command file (CMD).

**Example:** Instance 2 of MCL-Link has its corresponding MCLLNK02.INI file, where all settings defined by the user are stored for this instance.

Double clicking on the MCL-Link icon starts the program.



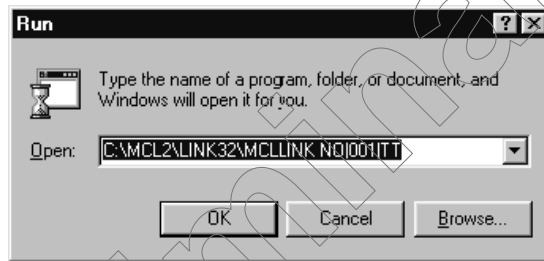
**Figure I-3. MCL-Link Window - Instance 2**

The number after the main window title shows the instance number of MCL-Link (in this case 02).

## Starting MCL-Link with Arguments

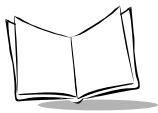
Commands can be sent to a terminal from a command line. The command is added to the MCL-Link program name as an argument:

1. Click on the **Start** button in the Task bar.
2. Select **Run**.
3. In the *Open* field, type the path and name of the MCL-Link program and the command (see Figure 1-4).
4. Each command must be sent as an argument so that it executes directly at MCL-Link start.



**Figure 1-4. Run MCL-Link with Arguments**

5. Click on the **OK** button to run the MCL-Link program.
6. The MCL-Link window displays (Figure 1-1).



## Communication Setup

To set up the host communication port, click on the **Setup** button in the MCL-Link window. The *MCL-Link Setup* window appears.



**Figure I-5. MCL-Link Setup Window**

To set up communications:

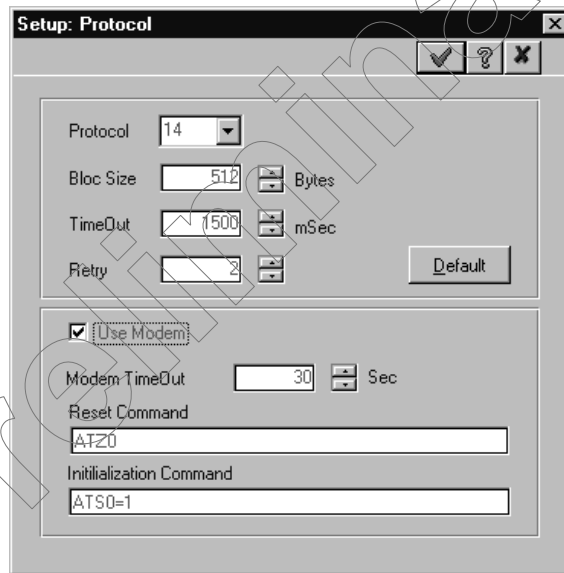
1. In the *Host ID* field, select the host identification number.
2. In the *Comm Port* list box, select the communication port that matches the one the terminal is plugged into on the host computer.
3. In the *Speed* list box, select the speed of the terminal communication port.
4. Click one of the *Log File* radio buttons to select the type of Log File to generate.

5. In the *Path for MCL Projects* field, select the directory where the MCL-Designer projects are located.
6. In the *General Path for Data Files* field, select the directory where the project data files are located.
7. Click on the OK button when done or the Cancel button to terminate the operation.

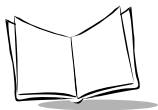
## Protocol

This dialog box defines how MCL-Link formats addresses and handles file transfer on the network. This applies only to the addresses formatted on the network.

Click on the Protocol button and the *Protocol* dialog box appears.



**Figure I-6. Setup: Protocol Dialog Box**

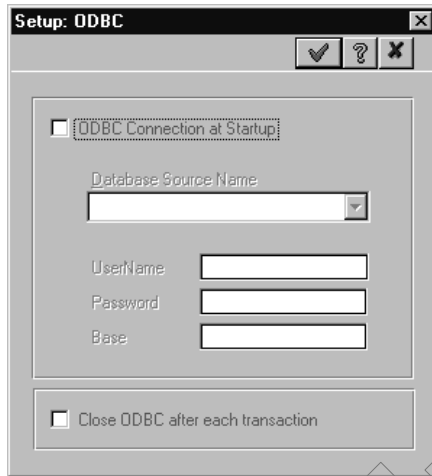


Field	Definition
Protocol	Defines the address format. The '0' indicates that decimal addressing is selected (001 to 099), and the '1' indicates that hexadecimal addressing is selected (001 to 254). The second digit defines the size of frame counter in a file transfer (0, 3 or 4 digits). Default value is 14.
Bloc Size	Maximum block size for transmitting frames on the network. (The maximum frame size, not a maximum size for data. The maximum value for this field is 512 characters. Allowed values are 64, 128, 256 or 512. Default value is 512.)
Time Out	Defines the time delay between retries.
Retry	Defines the number of retries.
Default	Resets all fields to their default values.
Use Modem	Sets a modem to use for remote connection.
Modem Time Out	Defines how many seconds to try the connection before timing out.
Reset Command	Defines the command to send when resetting the modem.
Initialization Command	Defines the command to send when initializing communications.

## ODBC

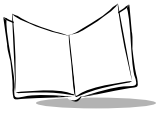
To access an SQL database through ODBC, the terminal sends a specific request («DR» data packet) that contains the MQD filename, the SQL query number and the data to MCL-Link. MCL-Link transfers this request to the MCL-Link ODBC/SQL engine that extracts the right SQL query. This query comes from a query definition file named MQD. The MQD contains all queries generated by MCL-Designer that allows you to access data in a specified database for a specific project. The MQD file is stored in the specified project directory. This directory is defined in the protocol setup. When the MCL-Link ODBC engine finds (or does not find) the corresponding data in the database, it sends a return code and data to the terminal through the MCL-Link. The return code indicates the number of records that are found in the database. Selecting the ODBC connection at startup automatically connects MCL-Link to the specified database.

Click on the ODBC button and the ODBC dialog box appears.



**Figure I-7. Setup: ODBC Dialog Box**

Field	Description
<b>ODBC Connection at Startup</b>	Sets the ODBC connection at startup. Deselect if not desired.
<b>Database Source Name</b>	Selects the database alias name as defined in the ODBC Data Source Administrator control panel. Choose from the pull-down menu.
<b>User Name</b>	Sends the user name if the database is password protected.
<b>Password</b>	Sends the password if the database is password protected.
<b>Base</b>	Reserved.
<b>Close ODBC after each transaction</b>	Closes the ODBC connection after each transaction. Deselect if not desired.

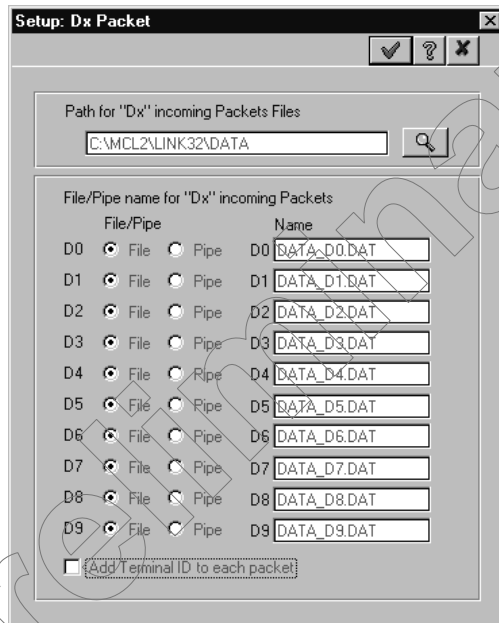


## Dx Packet

The Dx Packet dialog box allows the user to define the directory and the name of each file where data is saved when MCL-Link receives data records, or the name of the pipe.

When it is necessary to add the terminal ID into each data record received from the terminal, the Terminal ID is placed onto the three first digits of the record.

Click on the **Dx Packet** button and the *Dx Packet* dialog box appears.



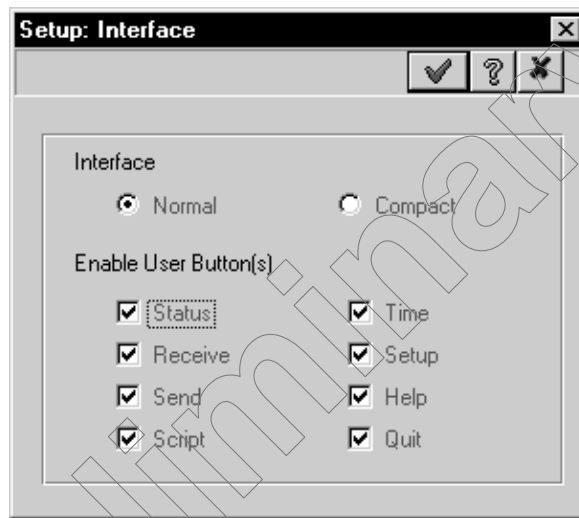
**Figure I-8. Setup: Dx Packet Dialog Box**

Field	Description
Path For “Dx” Incoming Packets Files	Routes the packets to the specified directory. Click the Browse button to change directories.
File/Pipe Name For “Dx” Incoming Packets	Assigns a file name to each individual incoming packet. Lists packets D0 through D9 and their corresponding file names or the name of the pipe.
Add Terminal ID To Each Packet	Attaches the Terminal ID number to the front of the record. Deselect if this is not required.

## Interface

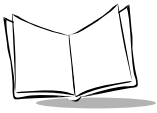
The Interface dialog box allows the user to enable or disable some or all buttons of the MCL-Link user interface.

Click on the *Interface* button and the *Interface* dialog box appears.



**Figure 1-9. Setup: Interface Dialog Box**

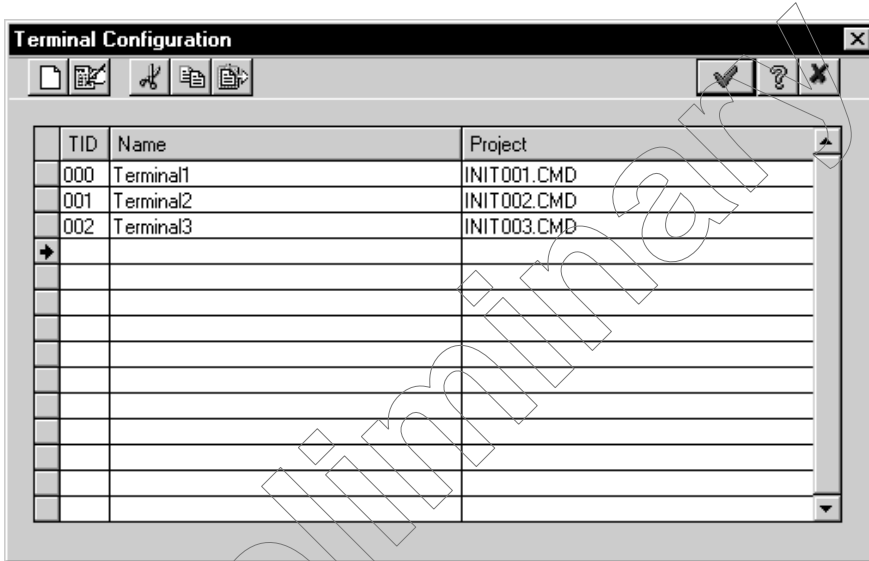
Field	Description
Interface	Determines the size of the window and buttons. When Compact Interface is chosen, only small buttons in a very small window appear on the screen.
Enable User Button(s)	Enables the checked buttons. Deselect the check boxes to disable the corresponding buttons.



## Terminal


An entire MCL project can be requested by the terminal (Auto-configuration). The user can define the project that is sent to the terminal.

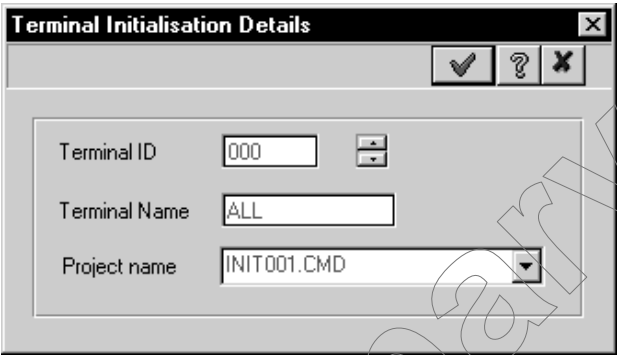
Click on the **Terminal** button and the *Terminal* dialog box appears.



**Figure I-10. Setup: Terminal Configuration Dialog Box**

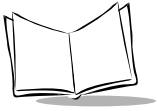
Field	Description
TID	Terminal Identification number - Identifies the terminal to receive the project.
Name	Confirms the name of the terminal to receive the project.
Project	States the file name of the project to send to the terminal.

To add information, click on the  button and the *Terminal Initialization Details* dialog box appears.



**Figure I-11. Setup: Terminal Initialization Details Dialog Box**

Field	Description
Terminal ID	Identifies the terminal to receive the project.
Terminal Name	Verifies the name of the terminal to receive the project.
Project Name	Identifies the name of the project to send to the terminal.



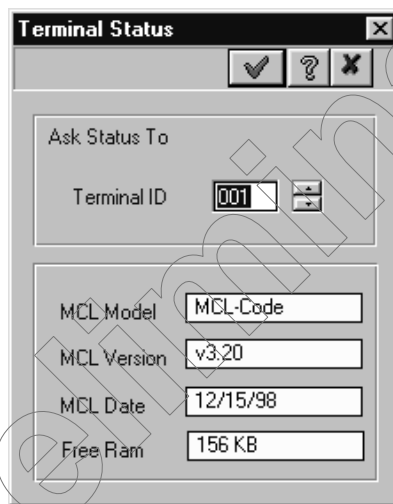
## Using MCL-Link

---

The MCL-Link main window contains buttons to perform all the different functions necessary for maintaining the terminals.

### Status

Click the Status button to initiate the status request operation for a specific terminal. The *Terminal Status* dialog box appears.

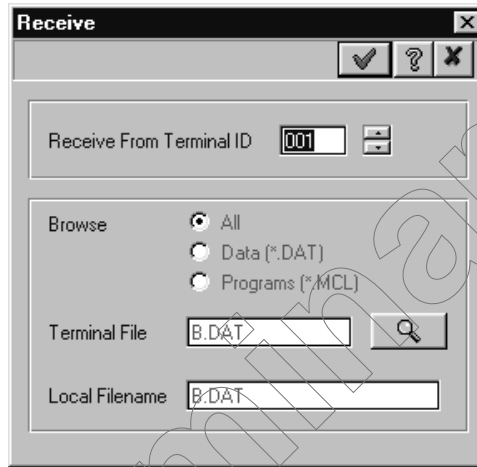


**Figure I-12. Terminal Status Window**

The terminal sends the version number of the MCL-Code program back to the host.

## Receive

Select the **Receive** button from the Link main window to start the File Receive operation from terminal and store it onto the host. The *Receive* dialog box appears.

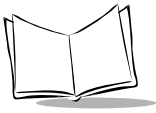


**Figure I-13. Receive Window**

The *Browse* field determines which files to search for.

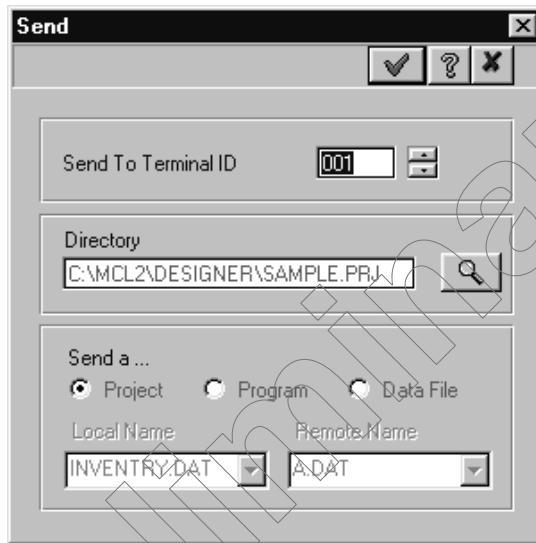
The *Magnification* icon requests a directory status from the terminal and allows the user to select the desired file to be uploaded.

The file received from the terminal is placed in the directory DATA with the name specified in the *Local Filename* field.



## Send

Selecting the Send button on the main window starts the File Send operation from computer to terminal. An entire project (programs and data files), programs only or data files only may be sent to the terminal. The *Send* dialog box appears.



**Figure 1-14. Send Window**

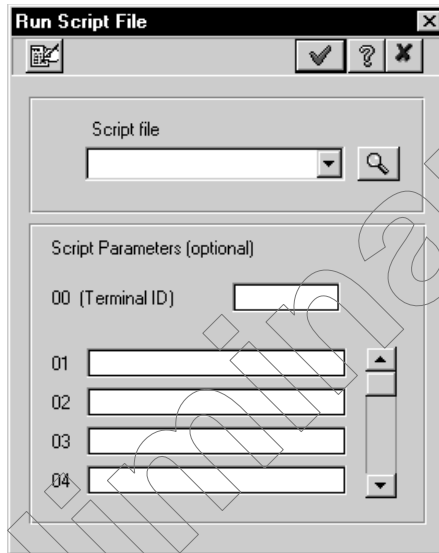
The **Magnification** button opens a local (computer) directory structure and allows the user to select another directory to locate the file.

The **Send a...** field determines what type of file to send to the terminal.

The **Local Name** field states the name of the file on the local computer, and **Remote Name** states the name that the file has on the terminal.


## Script

Selecting the **Script** button on the main window provides the ability to associate a script file (a list of commands) to a specified terminal. The *Script* dialog box appears.

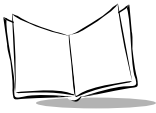


**Figure 1-15. Script Dialog Box**

The **Magnification** button next to the *Script file* field opens a command window on a local computer and allows the user to select a file (.CMD).

Click the  button to view the script file in a text editor such as Notepad. If a file is not selected using the **Magnification** button above, a window appears that allows the user to select the script file.

The **Script parameters** fields are optional. They are transferred to the script file as arguments. For example, &00 written in script file represents the terminal ID, &01 represents the script parameter 01, etc.



## Time

Selecting the Time button provides the ability to synchronize the terminal. The date and hour is updated.

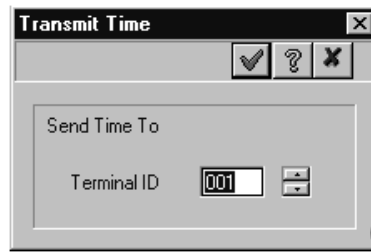


Figure I-16. Transmit Time Window

## About MCL-Link Files

Table 1-1 lists the files associated with MCL-Link.

Table 1-1. MCLLink File Descriptions

File	Description
MCLLINK.EXE	The executable program.
MCLLINK.HLP	The Windows Help file.
MCLLINK.INI	Contains the Setup information for MCL-Link.
MCLLINK.STA	Contains all status received from the terminals.
MCLLINK.ERR	Contains all transactions errors.
MCLLINK1.LOG MCLLINK2.LOG MCLLINK3.LOG	Log files.

When the size of MCLINK1.LOG file is approximately 50,000 bytes, it is copied into MCLLINK2.LOG which is itself copied into MCLLINK3.LOG. The data contained in the LOG files is dependent upon the parameter set in the SETUP screen (default is no Log file).

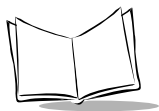
## MCL-Link Configuration File

The MCL-Link configuration file is named MCLLINK.INI. This is an ASCII file and can be edited with a common text editor. The values shown below are the default values.

### General Communication Parameters

#### [MCLLINK]

Host=099	Host address (valid entries: 1 to 254).
Port=1	Defines the used comm. Port (1 = COM1, 2 = COM2 ... )
Speed=38400	Defines the communication speed.
LogFile=1	Defines the log file mode. Valid entries are: 1 to 3.
LogSize=50000	Defines the max. size of the log file.
LogDate=ON	If ON current date is added to each log record.
LogTime=ON	If ON current time is added to each log record.
RecordLenMax=512	Defines the max size of each log record.
RepMCLDes=C:\MCL2\LINK32	Defines the MCL-Link path.
RepMCLDes1=C:\MCL2\DESIGNER\SAMPLE.PRJ	Defines the project path.
LastTO=001	
Minimize=OFF	Defines if MCL-Link starts in minimized mode.
WorkingMode=2	



## [PROTOCOL]

MODEL=PD3

BlocSize=512

Defines the maximum size of each communication frame.

Protocol=14

Defines the Address base system and the type of file transfer.

BroadCastDelay=1

TimeOut=1500

The maximum retry time-out (milliseconds).

Retry=2

Define the maximum number of retry.

Century=4

If equals 4 the date format is YYYY.

## Other MCL-Link Parameters

### [MODEM]

UseModem=0

Defines if MCL-Link uses a modem.

ModemReset=ATZ0

The modem initialization string.

ModemAnswer=ATS0=1

Defines if modem is set in auto-answer mode.

ModemTimeOut=30

Defines the maximum modem time-out (seconds).

### [USER\_BT]

Interface=1

Defines the interface type.

User\_Status=ON

Defines if interface button is visible.

User\_Receive=ON

Defines if interface button is visible.

User\_Send=ON

Defines if interface button is visible.

User\_Time=ON

Defines if interface button is visible.

User\_Setup=ON

Defines if interface button is visible.

User\_Help=ON

Defines if interface button is visible.

User\_Script=ON

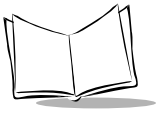
Defines if interface button is visible.

User\_Quit=ON

Defines if interface button is visible.

**[DATA]**

<b>Path=C:\MCL2\LINK32\DATA</b>	Defines the general path for data files.
<b>DxPath=C:\MCL2\LINK32\DATA</b>	Defines the path for files that stores data records D0..D9.
<b>DATA_D0=DATA_D0.DAT</b>	Defines the file name that stores data records D0.
<b>DATA_Q0=NO</b>	Defines if records D0 are stored in a file or in a data queue.
<b>DATA_D1=DATA_D1.DAT</b>	Defines the file name that stores data records D1.
<b>DATA_Q1=NO</b>	Defines if records D1 are stored in a file or in a data queue.
<b>DATA_D2=DATA_D2.DAT</b>	Defines the file name that stores data records D2.
<b>DATA_Q2=NO</b>	Defines if records D2 are stored in a file or in a data queue.
<b>DATA_D3=DATA_D3.DAT</b>	Defines the file name that stores data records D3.
<b>DATA_Q3=NO</b>	Defines if records D3 are stored in a file or in a data queue.
<b>DATA_D4=DATA_D4.DAT</b>	Defines the file name that stores data records D4.
<b>DATA_Q4=NO</b>	Defines if records D4 are stored in a file or in a data queue.
<b>DATA_D5=DATA_D5.DAT</b>	Defines the file name that stores data records D5.
<b>DATA_Q5=NO</b>	Defines if records D5 are stored in a file or in a data queue.
<b>DATA_D6=DATA_D6.DAT</b>	Defines the file name that stores data records D6.
<b>DATA_Q6=NO</b>	Defines if records D6 are stored in a file or in a data queue.
<b>DATA_D7=DATA_D7.DAT</b>	Defines the file name that stores data records D7.
<b>DATA_Q7=NO</b>	Defines if records D7 are stored in a file or in a data queue.
<b>DATA_D8=DATA_D8.DAT</b>	Defines the file name that stores data records D8.



<b>DATA_Q8=NO</b>	Defines if records D8 are stored in a file or in a data queue.
<b>DATA_D9=DATA_D9.DAT</b>	Defines the file name that stores data records D9.
<b>DATA_Q9=NO</b>	Defines if records D9 are stored in a file or in a data queue.
<b>AddTermID=OFF</b>	

### **[ODBC]**

<b>AutoConnect=ON</b>	Defines if Database is connected at MCL-Link Startup.
<b>Source=Sample Database</b>	The database source name.
<b>User=Forster</b>	The user name.
<b>PassWord=Jean_luc</b>	The user password.
<b>Base=MCLSAMP.MDB</b>	The base name of the database.

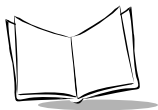
### **Configuration File Example**

```
[PROTOCOL]
MODEL=PD3
BlocSize=512
Protocol=14
BroadCastDelay=1
TimeOut=1500
Retry=2
Century=4
```

```
[MCLLINK]
Host=099
Port=1
Speed=38400
LogFile=1
LogSize=50000
LogDate=ON
LogTime=ON
RecordLenMax=512
RepMCLDes=C:\MCL2\LINK32
RepMCLDes1=C:\MCL2\DESIGNER\SAMPLE.PRJ
LastTO=001
Minimize=OFF
WorkingMode=2
```

```
[MODEM]
UseModem=0
ModemReset=ATZ0
ModemAnswer=ATS0=1
ModemTimeOut=30
```

```
[USER_BT]
Interface=1
User_Status=ON
User_Receive=ON
User_Send=ON
User_Time=ON
User_Setup=ON
User_Help=ON
User_Script=ON
User_Quit=ON
```



## MCL-Link Version 2.10 User's Guide

```
[DATA]
Path=C:\MCL2\LINK32\DATA
DxPath=C:\MCL2\LINK32\DATA
DATA_D0=DATA_D0.DAT
DATA_Q0=NO
DATA_D1=DATA_D1.DAT
DATA_Q1=NO
DATA_D2=DATA_D2.DAT
DATA_Q2=NO
DATA_D3=DATA_D3.DAT
DATA_Q3=NO
DATA_D4=DATA_D4.DAT
DATA_Q4=NO
DATA_D5=DATA_D5.DAT
DATA_Q5=NO
DATA_D6=DATA_D6.DAT
DATA_Q6=NO
DATA_D7=DATA_D7.DAT
DATA_Q7=NO
DATA_D8=DATA_D8.DAT
DATA_Q8=NO
DATA_D9=DATA_D9.DAT
DATA_Q9=NO
AddTermID=OFF

[SCRIPT]
ScriptTrace=OFF
LastSCR=C:\MCL2\LINK32\SCRIPT\MCL Samp1.CMD
=||||||||||||||
C:\MCL2\LINK32\SCRIPT\MCL Samp1.CMD=||||||||||||||02|

[DEBUG]
Trace=OFF
AllError=OFF

[DDE_SERVEUR]
DDE_READ=OFF

[ODBC]
AutoConnect=ON
Source=Sample Database
User=Forster
PassWord=Jean_luc
Base=MCLSAMP.MDB
```

```
[DEFAULT]
Send1=1
Send3=INVENTORY.DAT
Send4=A.DAT
Rece1=B.DAT
Rece2=B.DAT
```

## MCL Link Error File

The MCLLINK.ERR file contains all transactions errors. The first two letters identify the error, followed by the date, time, terminal ID, and command.

### Error When Initializing MCL-Link

CA Cancel by User.

DE Demo mode (no communication)

PO Open port comm. Error

Choose another comm. port or close the application that uses this comm. port (close the DOS Box if needed).

PI Initialize port comm. Error

Choose another comm. port or close the application that uses this comm. port (close the DOS Box if needed).

### Error In Command File

LB Label not found in a Command file.

Correct the Command file (refer to Chapter 2, *Command File*).

CM Bad or unknown command in a Command file.

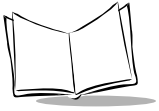
Correct the Command file (refer to Chapter 2, *Command File*).

### Error In Transaction

ID Bad terminal identification.

The command received from the terminal is incorrect. Correct the MCL program.

TO Time Out error, no response from the terminal or the modem.



**RJ** Reject received.

**MR** Max retry.

**XX** Unknown command received from the terminal.

The command received from the terminal is incorrect. Correct the MCL program.

Preliminary

## Error In Command When Files Are Implied

SS Sub-directory creation error.

The path of the file is incorrect.

FF File not found.

FN File Name incorrect.

FO File open error.

FS File error (bad structure, format, etc.).

FE File Operation error during copy, rename, append, delete.

Check file's existence, directory, disk space, etc.

## Error Code When Executing Another Program (EX or NO|xxx|EX)

01 Initialization error.

02 Program already finished.

03 Already in use.

04 Running too many programs (50 max).

05 Too much memory message (400 max).

06 Unknown destination.

07 No return message.

08 Not enough memory.

09 Unknown program or path.

10 Program is already running.

11 Initialization error.

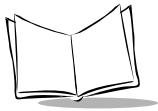
12 Memory allocation error.

13 DDE initialization error.

14 Error when connecting to the server.

15 DDE transaction error.

16 DDE received error.



## Error When ODBC is Used

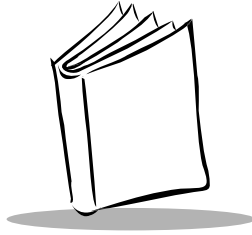
- O1 ODBC source not defined.
- O2 ODBC connection failed.
- O3 ODBC error on 16 bits systems when using ODBC 32 bits.
- O4 ODBC SQL command not defined.
- O5 ODBC command error.
- O6 ODBC error on file execution.

## Troubleshooting

---

If you encounter any problems:

- ◆ Check that the data is sent correctly from your host (terminal ID, filename,...).
- ◆ Check that power is correctly applied to the cradle or PIM.
- ◆ Check that your terminal is powered ON and in MCL-Link mode.
- ◆ Check communication parameters on the terminal and host computer.
- ◆ Parameters must be set the same on both the terminal and the host computer.
- ◆ Check the RS232 parameters on host side (connected to good communication port, etc.).
- ◆ Check your RS232 cable.



## Chapter 2

### Command File

#### Introduction

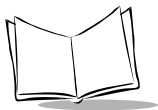
---

A command file can provide instruction to MCL-Link. This file contains a list of commands to execute. There can only be one command per line. The command file MCLLINK.CMD must be created in the current MCLLINK directory. This file is checked by MCLLINK.EXE several times every second. The commands are subdivided into REMOTE commands and LOCAL commands.

Table 2-1 lists the REMOTE commands that are sent from the host computer to the terminal.

**Table 2-1. Remote Commands**

Command	Description
NO xxx TM	Transmit Program
NO xxx TF	Transmit File
NO xxx FR	File Request
NO xxx TT	Transmit Time
NO xxx RZ	Reset
NO xxx ST	Status Request
NO xxx FC	File Copy
NO xxx FN	File Rename
NO xxx FA	File Append
<b>Note:</b> xxx is the terminal ID (from 001 to 254).	



**Table 2-1. Remote Commands (Continued)**

Command	Description
NO xxx FD	File Delete
NO xxx QX	Quit MCL-Link
<b>Note:</b> xxx is the terminal ID (from 001 to 254).	

Table 2-2 lists the LOCAL commands that are performed on the host computer.

**Table 2-2. Local Commands**

Command	Description
WT	Wait
QX	Quit
QT	Quit on Time Out
FN	File Rename
FC	File Copy
FD	File Delete
FA	File Append
EX	Execute the Program
SK	Skip
LB	Label or Mark
IF	Test and Branch
MD	MoDem Command
**	Comment

## Syntax Of Commands

---

| = Separator ASCII 124.

\ = Backslash character ASCII 92.

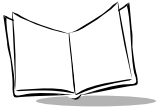
xxx = Terminal ID (range: 001 to 254).

Terminal\_File\_Name = The File Name in the terminal: a letter (from A to P) for data files, or a digit (from 0 to 10) for MCL programs.

PC\_File\_Name = The File Name on the PC Side. Contains the directory and file extension (if no directory is specified, the current one is used).

\\MCLLINK\DATA Default File Path.

.DAT for Data file and .MCL for MCL program file  
Default File Extension.



## Remote Commands

---

### **Transmit Program (TM)**

#### **Function**

Transmits the MCL program to the terminal.

#### **Syntax**

NO|xxx|TM|PC\_File\_Name|Terminal\_File\_Name

where:

xxx = Terminal ID (range: 001 to 254).

PC\_File\_Name = Full name that contains directory and file extension (if no directory is specified, the current one is used).

Terminal\_File\_Name = Represented by a digit (0 to 10) for MCL programs.

#### **Example**

NO|001|TM|C:\MCLD\PROJECT.PRJ\MAIN.MCL|0

## ***Transmit Data File (TF)***

### **Function**

Transmits a data file to the terminal.

### **Syntax**

NO|xxx|TF| *PC\_File\_Name* | *Terminal\_File\_Name*

where:

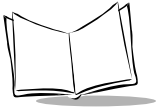
xxx = Terminal ID (range: 001 to 254).

PC\_File\_Name = Full name that contains directory and file extension (if no directory is specified, the current one is used).

Terminal\_File\_Name = Represented by a letter (from A to P) for data files.

### **Example**

NO|001|TF|C:\MCLD\PROJECT.PRJ\DATAFILE.DAT|A



## **Data File Request (FR)**

### **Function**

Requests a data file from the terminal.

### **Syntax**

NO|xxx|FR|*Terminal\_File\_Name*|*PC\_File\_Name*

where:

xxx = Terminal ID (range: 001 to 254).

PC\_File\_Name = Full name that contains directory and file extension (if no directory is specified, the current one is used).

Terminal\_File\_Name = represented by a letter (from A to P) for data files.

### **Example**

NO|001|FR|A|C:\MCLD\PROJECT.PRJ\DATAFILE.DAT

## ***Transmit Current Date and Time (TT)***

### **Function**

Transmits current time and date to the terminal.

### **Syntax**

NO|xxx|TT

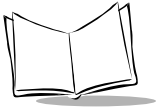
where:

xxx = Terminal ID (range: 001 to 254).

### **Example**

NO|001|TT

Preliminary



## Reset Terminal (RZ)

### Function

Resets the terminal.

### Syntax

NO|xxx|RZ|0

NO|xxx|RZ|1|x

NO|xxx|RZ|2

NO|xxx|RZ|2|Filename

NO|xxx|RZ|4

where:

xxx = Terminal ID (range: 001 to 254).

0 = warm re-boot.

1 = MCL program.

2 = all data files or A to P file.

4 = cold re-boot.

x = program number (0 through 10).

### Example

NO|001|RZ|2

## Status Request (SR)

### Function

Requests status from the terminal.

### Syntax

NO|xxx|SR|1

NO|xxx|SR|3|*File\_Name*

where:

xxx = Terminal ID (range: 001 to 254).

File\_Name = a file in the terminal (A through P) for data files.

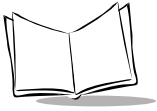
1 = terminal.

3 = a file.

### Example

NO|001|SR|3|A

The response to an SR command is an ST status transmit command. The file MCLLINK.STA contains all the status (ST) frames received from the terminal(s).



## **File Copy (FC)**

### **Function**

Copies one file on the terminal into another file on the terminal.

### **Syntax**

```
NO|xxx|FC|Old_file|New_file
```

where:

xxx = Terminal ID (range: 001 to 254).

Old\_file = name of the file to be copied.

New\_file = name of the file old\_file is copied into.

### **Example**

```
NO|001|FC|A|B
```

File **A** is copied to file **B**. Files **A** and **B** are identical after the copy.

## **File Rename (FN)**

### **Function**

Renames a file on the terminal.

### **Syntax**

NO|xxx|FN|*Old\_file*|*New\_file*

where:

xxx = Terminal ID (range: 001 to 254).

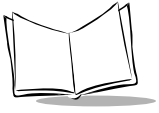
Old\_file = name of the file that is to be renamed.

New\_file = new name for file old\_file.

### **Example**

NO|001|FN|A|B

File A is renamed File B.



## **File Delete (FD)**

### **Function**

Deletes a file from the terminal.

### **Syntax**

NO|xxx|FD|*file\_Name*

where:

xxx = Terminal ID (range: 001 to 254).

file\_Name = name of the file to be deleted.

### **Example**

NO|001|FD|A

Preliminary

## ***File Append (FA)***

### **Function**

Adds a file to another file on the terminal.

### **Syntax**

NO|xxx|FA|File\_1|File\_2

where:

xxx = Terminal ID (range: 001 to 254).

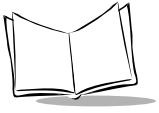
File\_1 = name of file to be added to File\_2.

File\_2 = name of file that File\_1 is added to.

### **Example**

NO|001|FA|A|B

File A is added to File B. File A remains unchanged.



## **Exit MCL-Link (QX)**

### **Function**

Exits MCL-Link program on the terminal.

### **Syntax**

NO|xxx|QX|1

where:

xxx = Terminal ID (range: 001 to 254).

1 = terminal.

### **Example**

NO|001|QX|1

Preliminary

## Local Commands

---

### **Wait (WT)**

#### **Function**

Adds a delay to the command file.

#### **Syntax**

**WT***|Time\_in\_seconds*

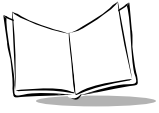
where:

Time\_in seconds = amount of time to wait.

#### **Example**

**WT**10

Wait 10 seconds before continuing to the next command.



## **Quit (QX)**

### **Function**

Closes the MCL-Link program.

### **Syntax**

QX

Preliminary

## **Quit on Time Out (QT)**

### **Function**

Closes the MCL-Link program after a set time of no activity.

### **Syntax**

`QT|Time_in_seconds`

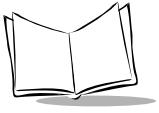
where:

Time\_in seconds = amount of time with no activity before closing the MCL-Link program.

### **Example**

`QT|10`

Close the MCL-Link program after 10 seconds of no activity with the terminal.



## **Label (LB)**

### **Function**

Defines a label in the command file.

### **Syntax**

**LB**|*Label*

where:

Label = name of the label.

### **Example**

**LB**|START

Defines the label *START* in the MCLLINK.CMD command file.

Preliminary

## ***Skip (SK)***

### **Function**

Goes to a label in the command file.

### **Syntax**

*SK|label*

where:

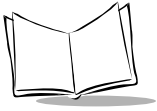
label = name of the label to go to or the number of lines above (-) or below (+) the current line.

### **Example**

*SK | START*

Go to the label *START* in the command file. The command Skip and Label are used together to make branches and loops in a command file.

*SK|+2*, *SK|-3* are used to jump directly to the corresponding number of lines.



## Test And Branch (IF)

### Function

Tests the value of a variable and then branches to a label upon condition.

### Syntax

```
IF|&99|=|0|Label_if_ok|Label_if_not_ok  
IF|&98|=|xxxx|Label_if_ok|Label_if_not_ok
```

where:

xxxx = value to test variable against.

Label\_if\_ok = label to go to if the condition is true.

Label\_if\_not\_ok = label to go to if the condition is false.

### Example

```
IF|&99|=|0|START|ERROR  
IF|&98|=|CONNECT|+1|ERROR
```

After each command, &99 contains "0" if the instruction had terminated correctly otherwise &99 contains "1".

```
LB|START  
NO|001|TF|A|DATA1.DAT  
IF|&99|=|0|+1|ERROR  
NO|001|TF|B|DATA2.DAT  
IF|&99|=|0|+1|ERROR  
WT|10  
SK|START  
LB|ERROR  
QT|10
```

## Modem Commands (MD)

### Function

Sends a modem command to the terminal.

### Syntax

MD|*Modem\_command*

where:

Modem\_command = modem command sent to the terminal.

### Example

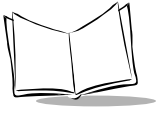
MD ATZ0	(reset the modem)
MD ATDT12345678	(dial a number)
MD ,+++,ATH0	(disconnect the line)

A comma in the modem command represents a 1 second delay.

Time out on MD commands is 30 seconds.

The MD Modem command sets &98 variable with the response of the modem (i.e., OK, 0, CONNECT, etc.)

```
LB|START
MD|ATDT123456787
IF|&98|!=|CONNECT|+1|ERROR
```



## **Comment (\*\*)**

### **Function**

Adds a comment to the command file.

No action is performed when the MCL-Link command file interpreter meets this line.

### **Syntax**

**\*\*|** *xxxx*

where:

*xxxx* = the comment.

### **Example**

**\*\*|** THIS IS A COMMENT

Preliminary

## **Local File Copy (FC)**

### **Function**

Copies one file into another file on the host computer.

### **Syntax**

`FC|Old_file|New_file`

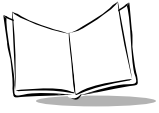
where:

Old\_file = name of the file to be copied.

New\_file = name of the file Old\_file is copied into.

### **Example**

`FC|ITEM.TXT|ITEM.BAK`



## **Local File Rename (FN)**

### **Function**

Renames a file on the host computer.

### **Syntax**

FN|*Old\_file*|*New\_file*

where:

Old\_file = name of the file that is to be renamed.

New\_file = new name for file old\_file.

### **Example**

FN|ITEM.TXT|ITEM.BAK

Preliminary

## **Local File Delete (FD)**

### **Function**

Deletes a file on the host computer.

### **Syntax**

FD|*file\_Name*

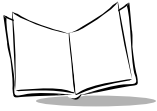
where:

file\_Name = name of the file to be deleted.

### **Example**

FD|ITEM.TXT

Preliminary



## **Local File Append (FA)**

### **Function**

Adds a file to another file.

### **Syntax**

`FA|File_1|File_2`

where:

File\_1 = name of file to be added to File\_2.

File\_2 = name of file that File\_1 is added to.

### **Example**

`FA|ITEM.TXT|ITEM.BAK`

The file ITEM.TXT is added to file ITEM.BAK. File ITEM.TXT remains unchanged.

## **Execute (EX)**

### **Function**

Executes a program on the host computer.

### **Syntax**

`EX|program_file_Name_and_Its_arguments`

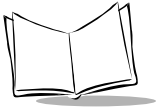
where:

`program_file_Name_and_Its_arguments` = name of program to run and any arguments.

### **Example**

`EX|NOTEPAD.EXE MyNote.DOC`

The program Notepad.exe is run and the argument MyNote.DOC is the file that opens.



## Commands Received from the Terminal

---

The following commands are MCL lines sent by the terminal to the host computer.

### **Receive Data Packet (D0-D9) from terminal**

#### **Function**

Sends data to appropriate data file.

#### **Syntax**

D0|data

where:

data = data added to the DATA\_DX.DAT file.

#### **Example**

NO|099|D0|19971010|231022|1111

D0 = Data is appended in DATA\_D0.DAT File in the general path.

D1 = Data is appended in DATA\_D1.DAT File in the general path.

...

D9 = Data is appended in DATA\_D9.DAT File in the general path.

The DATA\_Dx.DAT file is created if it doesn't exist.

## ***File Look-Up (CR) from terminal***

### **Function**

Looks for data in a file.

### **Syntax**

`CR|File_Name|Key_to_search`

where:

File\_Name = name of file to search.

Key\_to\_search = data to search for.

### **Example**

`NO|099|CR|PARTS.DAT|1234567890123`

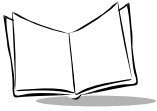
File\_Name is a text file sorted on a key, the key must be the first field in the file and the records length must be constant.

MCL-Link sends to the terminal a **CT** command with data from the record.

`CT|0` = key not found

`CT|1|Data_from_file` = key found, data are the rest of the line

`CT|9` = file not found



## **Transmit Remote Data File (TF) from terminal**

### **Function**

Transmits a data file to the host computer.

### **Syntax**

*TF|Terminal\_File\_Name|PC\_File\_Name*

where:

Terminal\_File\_Name = represented by a letter (from A to P) for data files.

PC\_File\_Name = Full name that contains directory and file extension (if no directory is specified, the current one is used).

### **Example**

NO|099|TF|A|C:\5CDATA\5CDATAFILE.DAT

## **Remote Data File Request (FR) from terminal**

### **Function**

Requests a data file from the host computer.

### **Syntax**

*FR|PC\_File\_Name|Terminal\_File\_Name*

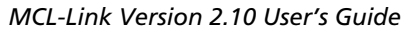
where:

Terminal\_File\_Name = Represented by a letter (from A to P) for data files.

PC\_File\_Name = Full name that contains directory and file extension (if no directory is specified, the current one is used).

### **Example**

NO|099|FR|C:\5CDATA\5CDATAFILE.DAT|A



## Function

## Syntax

where:

3 = a file.

PC\_File\_Name = a file on the host computer.

## NO|099|SR|3|C:\5CDATA\5CITEM.DAT

Request the status of the file C:\5CDATA\5CITEM.DAT.

## ***Host File Copy (FC) from terminal***

### **Function**

Copies a file on the host computer.

### **Syntax**

`FC|Old_file|New_file`

where:

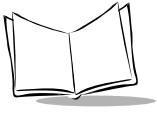
Old\_file = name of the file to be copied.

New\_file = name of the file Old\_file is copied into.

### **Example**

`NO|099|FC|ITEM.TXT|ITEM.BAK`

The file ITEM.TXT is copied to file ITEM.BAK. The two files are identical after the copy.



## **Host File Rename (FN) from terminal**

### **Function**

Renames a file on the host computer.

### **Syntax**

FN|Old\_file|New\_file

where:

Old\_file = name of the file that is to be renamed.

New\_file = new name for file Old\_file.

### **Example**

NO|099|FN|ITEM.TXT|ITEM.BAK

The file ITEM.TXT is renamed ITEM.BAK.

## ***Host File Delete (FD) from terminal***

### **Function**

Deletes a file from the host computer.

### **Syntax**

FD|*file\_Name*

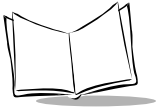
where:

file\_Name = name of the file to be deleted.

### **Example**

NO|099|FD|ITEM.TXT

Deletes the file ITEM.TXT from the host computer.



## **Host File Append (FA) from terminal**

### **Function**

Adds a file to another file on the host computer.

### **Syntax**

FA|File\_1|File\_2

where:

File\_1 = name of file to be added to File\_2.

File\_2 = name of file that File\_1 is added to.

### **Example**

```
NO|099|FA|ITEM.TXT|ITEM.BAK
```

The file ITEM.TXT is added to file ITEM.BAK. ITEM.TXT remains unchanged.

## ***Quit MCL-Link (QX) from terminal***

### **Function**

Closes the MCL-Link program on the host computer.

### **Syntax**

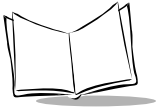
QX

### **Example**

NOI099IQX

Close the MCL-Link program.

Preliminary



## **Execute a Program (EX) from terminal**

### **Function**

Executes a program on the host computer.

### **Syntax**

`EX|program_file_Name_and_Its_arguments`

where:

`program_file_Name_and_Its_arguments` = name of program to run and any arguments.

### **Example**

`NO|099|EX|C:\5CNOTEPAD.EXE MyNote.DOC`

The program Notepad.exe is run and the argument MyNote.DOC is the file that opens.

## MCL-Link DLL

The MCL-Link provides a DLL interface to host applications in order to facilitate a client/server relationship between industrial terminals and a host application.

The main goal of this DLL is to receive operational terminal transaction data from the MCL-Link and send transaction-related data to a specific terminal via MCL-Link.

The DLL enables you to check if MCL-Link is running, and lets you start or stop MCL-Link on your Windows NT/95/98 environment.

Table 2-1 lists the DLL functions:

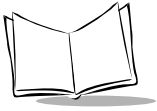
**Table 2-1. DLL Functions**

Function	Description
MCLLink_Start	Starts MCL-Link instance
MCLLink_Stop	Stops one MCL-Link instance
MCLLink_Check	Checks if MCL-Link is running
MCLLink_OpenPipe	Opens a named pipe
MCLLink_ClosePipe	Closes a named pipe
MCLLink_WaitData	Receives data from an MCL-Link instance
MCLLink_SendData	Sends data to a specified terminal
MCLLink_CheckTerminal	Checks the state of a specific terminal using an MCL-Link instance

### DLL Conventions

The following conventions are used:

- ◆ int and long represents a 32-bit signed integer (range -231 to 231- 1)
- ◆ char represents an 8 bit character (range 0 to 255)
- ◆ int \* represents a near pointer to an array of 32 bit signed integers
- ◆ char \* represents a near pointer to an array of characters.



## Return Code Standard Values

The return code is a 32-bit signed integer. A return code lower than 0 means that an error occurred. The nine significant error values are explained here:

-10	Error - Thread not found
-9	Too many processes/threads use the DLL (max 128)
-8	Error - Terminal not defined
-7	Error - Terminal not connected
-6	Error - Terminal connected but not reachable
-5	Error - A parameter is invalid
-2	Error - System error
-1	Error - Timeout
0	Error - MCL-Link not started

# MCL-Link DLL Functions

---

## ***MCLLink\_Start Function***

### **Function**

Starts MCL-Link instance (if not already started).

### **Syntax**

```
int MCLLink_Start(char *server, char *service, int arg)
```

where:

server = the name of the server for Windows NT only. Use «.» for Win95 and Win98 server.

service = the name of the service created by the MCL-Link instance.

**Example:** Instance 1 of MCL-Link creates a service named MCLLNK01, instance 2 creates service MCLLINK02, and so on to instance 8, which creates service MCLLNK08. When MCL-Link is started in single instance, the name of the service is MCLLINK.

*Interface mode:*

0 = normal window

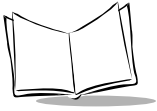
1 = maximized window

2 = minimized window

### **Return Values**

A return code greater than 0 indicates that MCL-Link instance is started correctly.

DLL function ordinal number: 2



## MCLLink\_Stop Function

### Function

Stops one MCL-Link instance.

### Syntax

```
int MCLLink_Stop(char *server, char *service)
```

where:

server = the name of the server for Windows NT only. Use «.» for Win95 and Win98 server.

service = the name of the service created by the MCL-Link instance.

**Example:** Instance 1 of MCL-Link creates a service named MCLLNK01, instance 2 creates service MCLLINK02, and so on to instance 8, which creates service MCLLNK08. When MCL-Link is started in single instance, the name of the service is MCLLINK.

### Return Values

A return code greater than 0 indicates that MCL-Link is stopped correctly.

DLL function ordinal number: 3

## MCLLink\_Check Function

### Function

Checks if MCL-Link is running.

### Syntax

```
int MCLLink_Check(char *server, char *service)
```

where:

server = the name of the server for Windows NT only. Use «.» for Win95 and Win98 server.

service = the name of the service created by the MCL-Link instance.

**Example:** Instance 1 of MCL-Link creates a service named MCLLNK01, instance 2 creates service MCLLINK02, and so on to instance 8, which creates service MCLLNK08. When MCL-Link is started in single instance, the name of the service is MCLLINK.

### Return Values

A return code greater than 0 indicates that MCL-Link is running.

A return code < = 0 indicates that MCL-Link is not running.

### Example

*Win95/Win98:*

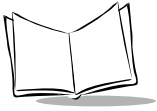
```
MCLLink_Check(«.», MCLLINK01)
```

*Windows NT:*

```
MCLLink_Check(«SERVER», «MCLLINK01»)
```

### Remark

DLL function ordinal number: 1



## MCLLink\_OpenPipe Function

### Function

Opens a named pipe.

### Syntax

```
int MCLLink_OpenPipe (char *server, char *service, char * buffer )
```

where:

server = the name of the server for Windows NT only. Use «.» for Win95 and Win98 server.

service = the name of the service created by the MCL-Link instance.

**Example:** Instance 1 of MCL-Link creates a service named MCLLNK01, instance 2 creates service MCLLINK02, and so on to instance 8, which creates service MCLLNK08. When MCL-Link is started in single instance, the name of the service is MCLLINK.

buffer = The name of the pipe (zero terminated string)

### Return Value

A return code greater than 0 indicates that has no error to open the specified pipe.

DLL function ordinal number: 9

### Note

The name of the pipe must be defined in the dispatcher and MCL-Link must be started. The name of the pipe may not exceed 60 bytes.

## **MCLLink\_ClosePipe Function**

### **Function**

Close a named pipe.

### **Syntax**

```
int MCLLink_ClosePipe (char *server, char *service, char * buffer )
```

where:

server = the name of the server for Windows NT only. Use «.» for Win95 and Win98 server.

service = the name of the service created by the MCL-Link instance.

**Example:** Instance 1 of MCL-Link creates a service named MCLLNK01, instance 2 creates service MCLLINK02, and so on to instance 8, which creates service MCLLNK08. When MCL-Link is started in single instance, the name of the service is MCLLINK.

buffer = The name of the pipe (zero terminated string)

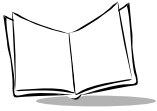
### **Return Value**

A return code greater than 0 indicates that has no error to close the named pipe.

DLL function ordinal number: 10

### **Note**

The name of the pipe may not exceed 60 bytes.



## **MCLLink\_WaitData Function**

### **Function**

Receives data from a MCL-Link instance.

### **Syntax**

```
int MCLLink_WaitData (char *server,  
                      char *service,  
                      int * Term,  
                      long Timeout,  
                      char * Cmd,  
                      char * Buffer,  
                      int Maxlen)
```

where:

server = the name of the server for Windows NT only. Use «.» for Win95 and Win98 server.

service = the name of the service created by the MCL-Link instance.

**Example:** Instance 1 of MCL-Link creates a service named MCLLNK01, instance 2 creates service MCLLNK02, and so on to instance 8, which creates service MCLLNK08. When MCL-Link is started in single instance, the name of the service is MCLLINK.

Term = terminal number (range: 1 to 254).

Timeout = timeout in msec.

Cmd = the pipe name (defined in the Dx Packet setup).

Buffer = data input receive buffer.

Maxlen = maximum length of input buffer.

### **Return Values**

A return code greater than 0 indicates that data has been received from the terminal which is connected and reachable.

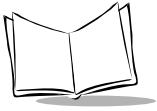
The return code, if greater than 0, indicates the number of received characters.

DLL function ordinal number: 6

## **Notes**

1. Buffer that receives data must be declared to the effective maximum data length +1 because a 0 (hexadecimal) is added as terminator.
2. The variable Term is written with the terminal number after the execution.
3. Received command (Packet Type) is copied in Cmd after the execution of the function.

Preliminary



## **MCLLink\_SendData Function**

### **Function**

Sends data to a specified terminal.

### **Syntax**

```
int MCLLink_SendData (char *server,  
                      char *service,  
                      int Term,  
                      long Timeout,  
                      char * Cmd,  
                      char * buffer,  
                      int Len)
```

where:

server = the name of the server for Windows NT only. Use «.» for Win95 and Win98 server.

service = the name of the service created by the MCL-Link instance.

**Example:** Instance 1 of MCL-Link creates a service named MCLLNK01, instance 2 creates service MCLLINK02, and so on to instance 8, which creates service MCLLNK08. When MCL-Link is started in single instance, the name of the service is MCLLINK.

Term = terminal number (range: 1 to 254).

Timeout = timeout in msec.

Cmd = command to send to the specified terminal (2 char.).

Buffer = data to send.

Len = length of data.

### **Return Values**

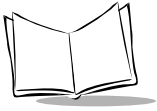
A return code greater than 0 means that data has been sent correctly to the terminal which is connected and reachable.

DLL function ordinal number: 7

**Note**

Timeout must be long enough in case of file or MCL-Code programs transfer to a specified terminal. If Timeout is set to 0, the function MCLLink\_SendData is not waiting for acknowledgement from the terminal. This command exits immediately.

Preliminary



## **MCLLink\_CheckTerminal Function**

### **Function**

Checks the state of a specific terminal using a MCL-Link instance.

### **Syntax**

```
int MCLLink_CheckTerminal(char *server, char *service, int Term)
```

where:

server = the name of the server for Windows NT only. Use «.» for Win95 and Win98 server.

service = the name of the service created by the MCL-Link instance.

**Example:** Instance 1 of MCL-Link creates a service named MCLLNK01, instance 2 creates service MCLLINK02, and so on to instance 8, which creates service MCLLNK08. When MCL-Link is started in single instance, the name of the service is MCLLINK.

Term = terminal number (range: 1 to 254).

### **Return Values**

A return code greater than 0 indicates that the terminal is connected.

DLL function ordinal number: 4



## Index

### C

command syntax	2-3
commands from terminal	
Data File Request	2-31
Execute	2-38
File Append	2-36
File Copy	2-33
File Delete	2-35
File Look-Up	2-29
File Rename	2-34
Quit	2-37
Receive Data Packet	2-28
Status Request	2-32
Transmit Data File	2-30
communication setup	1-8
configuration file	1-21

### D

DLL conventions	2-39
DLL Functions	
MCLLink_Check	2-43
MCLLink_CheckTerminal	2-50
MCLLink_ClosePipe	2-45
MCLLink_OpenPipe	2-44
MCLLink_SendData	2-48
MCLLink_Start	2-41
MCLLink_Stop	2-42
MCLLink_WaitData	2-46

### E

error messages	1-27
----------------	------

### I

information, service	viii
installation	1-2
Introduction	1-1

### L

local commands	2-1, 2-2
Comment	2-22
Execute	2-27
File Append	2-26
File Copy	2-23
File Delete	2-25
File Rename	2-24
Label	2-18
Modem Commands	2-21
Quit	2-16
Quit on Time Out	2-17
Skip	2-19
Test and Branch	2-20
Wait	2-15

### M

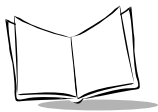
MCL Collection	1-2
MCL-Link DLL	2-39
MCL-Link files	1-20

### O

operating modes	1-3
-----------------	-----

### R

remote commands	2-1
Data File Request	2-6



Exit MCL-Link .....	2-14
File Append .....	2-13
File Copy .....	2-10
File Delete .....	2-12
File Rename .....	2-11
Reset Terminal .....	2-8
Status Request .....	2-9
Transmit Current Date and Time ....	2-7
Transmit Data File .....	2-5
Transmit Program .....	2-4
Return Code Standard Values .....	2-40

## **S**

service information .....	viii
starting MCL-Link .....	1-4
starting MCL-Link with arguments .....	1-7
symbol support center .....	ix

## **T**

transaction errors .....	1-27
troubleshooting .....	1-30

## **U**

using MCL-Link .....	1-16
----------------------	------

Preliminary

# Tell Us What You Think...

We'd like to know what you think about this Manual. Please take a moment to fill out this questionnaire and fax this form to: (516) 738-3318, or mail to:

Symbol Technologies, Inc.  
One Symbol Plaza M/S B-4  
Holtsville, NY 11742-1300  
Attn: Technical Publications Manager

**IMPORTANT:** If you need product support, please call the appropriate customer support number provided. Unfortunately, we cannot provide customer support at the fax number above.

User's Manual Title: \_\_\_\_\_  
(please include revision level)

How familiar were you with this product before using this manual?

☐ Very familiar    ☐ Slightly familiar    ☐ Not at all familiar

Did this manual meet your needs? If not, please explain. \_\_\_\_\_

What topics need to be added to the index, if applicable? \_\_\_\_\_

What topics do you feel need to be better discussed? Please be specific. \_\_\_\_\_

What can we do to further improve our manuals? \_\_\_\_\_

Thank you for your input—We value your comments.